



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/772,207	02/03/2004	Sebastian Lange	MS1-1823US	4217
22801 7590 08/04/2009				
LEE & HAYES, PLLC 601 W. RIVERSIDE AVENUE SUITE 1400 SPOKANE, WA 99201				
EXAMINER				
DOAN, TRANG T				
ART UNIT		PAPER NUMBER		
2431				
MAIL DATE		DELIVERY MODE		
08/04/2009		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/772,207

Applicant(s)

LANGE ET AL.

Examiner

TRANG DOAN

Art Unit

2431

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 06 April 2009.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-9, 11-15, 17-22, 24-34, 36-43, 45 and 51 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-9, 11-15, 17-22, 24-34, 36-43, 45 and 51 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 02/23/2004 is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-849)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

1. This action is in response to the amendment filed on 04/06/2009.
2. Claims 1, 12, 19-26 and 38 have been amended.
3. Claims 10, 16, 23, 35, 44 and 46-50 have been canceled.
4. Claims 1-9, 11-15, 17-22, 24-34, 36-43, 45 and 51 are pending for consideration.

Response to Arguments

5. Applicant's argument with respect to the 35 U.S.C. 101 and 112, 2nd paragraph, rejection has been fully considered in view of the amendment filed on 04/06/2009, which has been made in record, and the 35 U.S.C. 101 and 112, 2nd paragraph, rejection have been withdrawn.
6. Applicant's arguments with respect to claims 1-9, 11-15, 17-22, 24-34, 36-43, 45 and 51 have been considered but are moot in view of the new ground(s) of rejection.

Claim Objections

7. Claim 6 is objected to because of the following informalities: the limitation "an cross assembly call" should be changed to "a cross assembly call". Appropriate correction is required.

Claim Rejections - 35 USC § 103

8. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

9. Claims 1-9, 11-15, 17-22, 24-34, 36-43, 45 and 51 are rejected under 35 U.S.C. 103(a) as being unpatentable over Koved et al. (Reference U: Access Rights Analysis for Java) (hereinafter Koved) in view of Wong et al. (Reference V: Securing Programming with .NET) (hereinafter Wong).

Regarding claim 1, Koved discloses simulating the execution of all calls from an assembly to another assembly for all execution paths of one or more assemblies in the managed code, wherein the assembly comprises one or more files versioned and deployed as a unit, wherein the managed code is a managed shared library or an executable (**Koved: on page 1, column 2, under INTRODUCTION heading, second paragraph: “developer reads ... libraries used (including the Java run-time libraries”) and reduces the required access rights**), wherein all managed code is contained within the one or more assemblies, wherein the execution of each assembly is statically simulated without actually running a corresponding managed code to simulate all possible calls and corresponding flow of argument data (**Koved: page 1 see Abstract section: test cases...cover all paths and on page 2, column 2**); finding a set of required permissions for each execution path by one or more simulated

stack walks that each include a plurality of the assemblies, wherein each call in each execution path has a corresponding said permissions set, wherein each assembly has one or more execution paths representing a different data and a control flow, **(Koved: on page 2, column 2 and last paragraph: “a formalization of stack introspection, which examines authorization based on the principals (signers and/or origin of the code) currently active in a Thread stack at run-time, as is found in Java ... at each stack frame, as well as the result of any authorization test encountered” and page 3, column 1 and first paragraph: “the runtime stack ... to discover authorization requirements by analyzing all possible paths through the program”);** and deriving the security requirements for execution paths corresponding to the one or more assemblies by using the union of the gathered permission sets across the execution paths corresponding to the one or more assemblies, wherein the union estimates the security requirements that will be triggered against the one or more assemblies during the actual execution of the one or more assemblies and whether a security exception will be triggered during the actual execution **(Koved: see page 3 column 1 first paragraph and page 4 column 1 paragraphs 1-3: determine access rights required by Java programs or libraries).**

Koved does not explicitly disclose wherein the simulated stack walk comprises: entering an execution path corresponding to a static simulation of execution of the assembly; entering a public entry point of a method in the assembly; gathering a permission set for the method in the assembly; determining whether the method in the assembly calls another method in the assembly or in an another assembly; gathering a

permission set for the another method called by the method in the assembly; and creating a union of the gathered permission sets.

However, Wong discloses wherein the simulated stack walk comprises: entering an execution path corresponding to a static simulation of execution of the assembly (**Wong: see page 2 under Stack Waking section**); entering a public entry point of a method in the assembly (**Wong: see page 2 under Stack Waking section**); gathering a permission set for the method in the assembly (**Wong: see page 3 under Requesting minimum required permissions for execution and Securing sensitive information section**); determining whether the method in the assembly calls another method in the assembly or in an another assembly (**Wong: see page 2 under Stack Waking section**); gathering a permission set for the another method called by the method in the assembly (**Wong: see page 3 under Requesting minimum required permissions for execution and Securing sensitive information section**); and creating a union of the gathered permission sets (**Wong: see page 2 under Stack Waking section: stack walk...walking through all records in the call chain and determining if they have appropriate access rights and page 3 under Requesting minimum required permissions for execution and Securing sensitive information section: .NET application programmer can explicitly specify the minimum level of required permissions for the code to execute**). Therefore, it would have been obvious to a person skilled art at the time the invention was made to have included in Koved the feature of Wong as discussed above to build secure application by creating an environment in which applications can be controlled and security decisions made

based on the evidence of the code, the role of the user, and the security policy of the environment (**Wong: see page 4 under Conclusion section**).

Regarding claim 2, Koved as modified discloses wherein the execution paths for only one said assembly in managed code are simulated to find the set of required permissions for each said execution path by a union of the permissions for each said execution path (**Koved: on page 2, column 2, third paragraph; page 3, column 1, first paragraph; and page 3 and page 4, under Authorization Model section: “In this paper...an invocation graph and data flow analysis...more accurate authorization information.” “Our approach...discover authorization requirements by analyzing all possible paths through the program.” “It can be seen...the value of Required Permissions (n) (i.e., $RP(n)$) at the input to a node n...by means of a set of union operation”**).

Regarding claim 3, Koved as modified discloses wherein: the one or more assemblies in managed code correspond to an application (**Koved: page 3, column 1, third paragraph: “Each Java application class...associated with a set of right, or privileges, granted to the code**); and the set of required permissions for each said execution path comprises a union of the permissions for each said execution path (**Koved: page 3, column 2; and page 4 column 2: “Since, in general...along paths towards nodes in N_{start} . This process associates a set of required requirements $RP(n)$ with each node n in N_{start} . More precisely, it computes $RP(n)$ for all n**

belong to N". "It can be seen that the data flow...by means of a set union operation").

Regarding claim 4, Koved as modified discloses wherein: the assemblies in managed code correspond to a shared library (**Koved: page 8, column 1, third paragraph: "For a given application or classes in a library...identify the set of Java 2 Permissions required for each class in the analysis scope"**); and the set of required permissions for each said execution path comprises one separate permission set per entry point in the shared library (**Koved: on page 1, under ABSTRACT section; and page 2, column 1, under Prior Work section: "This paper presents...compute at each program point the set of access rights required by the code" ... "authorization tests...to the current approach to defining authorization points within code")**).

Regarding claim 5, this claim has limitations that are similar to those of claims 1-2 and 3, thus it is rejected with the same rationale applied against claims 1-2 and 3 above.

Regarding claim 6, Koved as modified discloses wherein one of more of the calls in at least one said execution path is a cross assembly call (**Koved: on page 2, column 2, third paragraph: "In the aforementioned works...Java runtime calls one of the**

SecurityManager authorization methods...to correctly identify authorization requirements”).

Regarding claim 7, Koved as modified discloses wherein: the managed code is built to make use of a common language runtime (**Koved: see page 2, column 2, third paragraph: “In the aforementioned works...Java runtime calls one of the SecurityManager authorization methods...to correctly identify authorization requirements”**); each said assembly is packaged as an executable entity or as a data link library entity and each said assembly includes one or more methods (**Koved: on page 1, under ABSTRACT section; and page 7, column 2, second and third paragraph: “The tool...to identify the access rights requirements for the product to enable it to run using Java 2 security model”**).

Regarding claim 8, Koved as modified discloses wherein the simulation of the execution of each said execution path comprises a simulation of the flow of argument data using intra and extra method data flow analysis for each said method (**Koved: on page 2, column 1, second and third paragraph; and page 6, column 2, second paragraph: “To summarize...We present a context sensitive, flow sensitive analysis for computing the access rights requirements of a program.” “To minimize conservativeness...the order of execution of instructions both intra- and inter procedurally thus improving the accuracy of the resulting graph”**).

Regarding claim 9, Koved as modified discloses wherein when the executable has permissions to execute that are not less than a union of permission sets for each said execution path, any dynamic execution of the executable will not trigger a security exception (**Koved: on page 3, under Authorization Model-Access Rights Invocation Graph section; page 7, column 1, second paragraph; and page 8, column 1, first paragraph: “Performance is improved...NullPointerException in package...”**).

Regarding claim 11, Koved as modified discloses a computer readable storage medium having a tangible component including machine readable instructions for implementing the method as defined in Claim 1 (**Koved: on page 4, column 1, third paragraph**).

Regarding claim 12, this claim has limitations that are similar to those of claims 1-6, thus it is rejected with the same rationale applied against claims 1-6 above.

Regarding claim 13, Koved as modified discloses wherein the managed code environment comprises: a managed code portion including: the assemblies (**Koved: see ABSTRACT section**); and a virtual machine (**Koved: on page 3, column 1, paragraph 3: “Each Java application...the Java Virtual Machine...to the code”**); a native code portion including: an execution engine for the virtual machine (**Koved: see ABSTRACT section: “Java...protects systems...execute the code...in deployed systems”**); and an operating system under the execution engine (**Koved:**

see ABSTRACT section: “Java...protects systems...execute the code...in deployed systems”).

Regarding claim 14, this claim has limitations that are similar to those of claim 7, thus it is rejected with the same rationale applied against claim 7 above.

Regarding claim 15, this claim has limitations that are similar to those of claim 9, thus it is rejected with the same rationale applied against claim 9 above.

Regarding claim 17, Koved as modified discloses wherein the managed code environment enforces partial trust security contexts (**Koved: on page 3, column 1, 2 paragraph: “Rather than analyzing...enforce specific security policies...updated to enable the code to execute”).**

Regarding claim 18, this claim has limitations that are similar to those of claim 11, thus it is rejected with the same rationale applied against claim 11 above.

Regarding claim 19, this claim has limitations that are similar to those of claim 1, thus it is rejected with the same rationale applied against claim 1 above.

Regarding claim 20, this claim has limitations that are similar to those of claim 7 and 14, thus it is rejected with the same rationale applied against claims 7 and 14 above.

Regarding claim 21, this claim has limitations that are similar to those of claim 13, thus it is rejected with the same rationale applied against claim 13 above.

Regarding claim 22, this claim has limitations that are similar to those of claim 20, thus it is rejected with the same rationale applied against claim 20 above.

Regarding claim 24, this claim has limitations that are similar to those of claim 16, thus it is rejected with the same rationale applied against claim 16 above.

Regarding claim 25, this claim has limitations that are similar to those of claim 17, thus it is rejected with the same rationale applied against claim 17 above.

Regarding claim 26, this claim has limitations that are similar to those of claim 1, thus it is rejected with the same rationale applied against claim 1 above.

Regarding claim 27, Koved as modified discloses means for compiling the assemblies from an intermediate language code and metadata into native code; and means for loading the native code with a Common Language Runtime loader in the

native code portion to load the compiled native code, wherein the execution engine means executes the compiled native code in the native code portion (**Koved: on page 3, column 1, first paragraph: "Permission.implies...test cases")**).

Regarding claim 28, Koved as modified discloses wherein the managed code portion further comprises one or more files associated with user code that, when compiled into an intermediate language code and metadata generated by a language compiler, are represented by the assemblies (**Koved: on page 3, column 1, third paragraph: "Each Java application class...Java Virtual Machine...privileges, granted to the code")**).

Regarding claim 29, Koved as modified discloses wherein the execution engine means in the native code portion further comprises a compiler to compile each said assembly into native code for execution by the native code portion (**Koved: on page 3, column 1, third paragraph: "Each Java application class...Java Virtual Machine...privileges, granted to the code")**).

Regarding claim 30, Koved as modified discloses wherein the execution engine means in the native code portion further comprises: a Just In Time compiler to compile each said assembly into native code; and a common language runtime loader to load the compiled native code for execution by the native code portion (**on page 3, column**

1, third paragraph: “Each Java application class...Java Virtual Machine...privileges, granted to the code”).

Regarding claim 31, Koved as modified discloses means, in the native code portion, for forming a response to the call; and means for returning the response to the first assembly in the managed code portion (**Koved: on page 3, column 1, third paragraph; and page 4, column 1, first paragraph: “Each Java application class...Java Virtual Machine...privileges, granted to the code”).**

Regarding claim 32, Koved as modified discloses wherein: the managed code is built to make use of a common language runtime; each said assembly is packaged as an executable entity or as a data link library entity; and each said assembly includes one or more methods (**Koved: on page 1, under ABSTRACT section; and page 7, column 2, second and third paragraph: “The tool...to identify the access rights requirements for the product to enable it to run using Java 2 security model”).**

Regarding claim 33, Koved as modified discloses wherein the simulation of the execution comprises, for each said execution path, a simulation of the flow of argument data using intra and extra data flow analysis for each said method (**Koved: on page 2, column 1, second and third paragraph; and page 6, column 2, second paragraph: “To summarize...We present a context sensitive, flow sensitive analysis for computing the access rights requirements of a program.” “To minimize**

conservativeness...the order of execution of instructions both intra- and inter procedurally thus improving the accuracy of the resulting graph”).

Regarding claim 34, this claim has limitations that are similar to those of claim 9, thus it is rejected with the same rationale applied against claim 9 above.

Regarding claim 36, Koved as modified discloses wherein each call in each simulated stack walk has a corresponding permissions set (**Koved: on page 3 under Authorization Model-Access Rights Invocation Graph section: “For any node...set of required Permissions for n”).**

Regarding claim 37, this claim has limitations that are similar to those of claim 17, thus it is rejected with the same rationale applied against claim 17 above.

Regarding claim 38, this claim has limitations that are similar to those of claims 1 and 26, thus it is rejected with the same rationale applied against claims 1 and 26 above.

Regarding claim 39, this claim has limitations that are similar to those of claims 27 and 28, thus it is rejected with the same rationale applied against claims 27 and 28 above.

Regarding claim 40, this claim has limitations that are similar to those of claim 30, thus it is rejected with the same rationale applied against claim 30 above.

Regarding claim 41, this claim has limitations that are similar to those of claim 22, thus it is rejected with the same rationale applied against claim 22 above.

Regarding claim 42, this claim has limitations that are similar to those of claim 8, thus it is rejected with the same rationale applied against claim 8 above.

Regarding claim 43, this claim has limitations that are similar to those of claims 9 and 15, thus it is rejected with the same rationale applied against claims 9 and 15 above.

Regarding claim 45, this claim has limitations that are similar to those of claim 17, thus it is rejected with the same rationale applied against claim 17 above.

Regarding claim 51, Koved as modified discloses wherein the union of the permission sets separately identifies a permission set for each public entry point of the library (**Koved: on page 2, column 2, third paragraph; page 3, column 1, first paragraph; and page 3 and page 4, under Authorization Model section: “In this paper...an invocation graph and data flow analysis...more accurate authorization information.” “Our approach...discover authorization requirements by analyzing**

all possible paths through the program.” “It can be seen...the value of Required Permissions (n) (i.e., RP(n)) at the input to a node n...by means of a set of union operation”).

Conclusion

10. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to **TRANG DOAN** whose telephone number is (571)272-0740. The examiner can normally be reached on Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William R. Korzuch can be reached on (571) 272-7589. The fax phone

Art Unit: 2431

number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Trang Doan/
Examiner, Art Unit 2431

/Christopher A. Revak/

Primary Examiner, Art Unit 2431